1. 每个概念被创造都有其意义，请简述“进程”这个概念在Linux系统中有什么用途。

进程是描述程序执行过程和资源共享的基本单位，所有的控制和管理程序运行的东西都需要使用一个特殊的数据结构来存储它、保存他、维护它和管理它，这就是进程。其主要目的是控制和协调程序的执行。

1. 代码段A执行后会产生25个进程，而代码段B执行产生的进程有5个。这是因为每调 用fork()函数系统会复制原程序，在代码段A中当第一次调用fork()函数后会有两个进程，这两个进程继续进行接下来的循坏，即父进程与其创建的子进程都会进行第二次循环，并且每一次循环后进程的总数都为当前进程总数的两倍，5次循环后，进程总数为25个；在代码段B中由于进行判断了是否为父进程的判断，每当调用fork()函数产生两个进程，其中当子进程执行时会break而跳出循环，即代码段B每调用一次fork()函数产生两个进程，但是只有一个进程可以继续执行循环，5次循环后，进程总数为6个，其中6个子进程。
2. 用自己的话阐述什么是僵尸进程，并描述进程通过调用 wait() 捕获僵尸态的子进程的过程。

由于进程的执行顺序一般都是随机的，当一个父进程创建了子进程后，子进程先结束，并且父进程没有处理回收子进程，此时的子进程就成为了僵尸进程，即一个已经终止了但还未被回收的进程。僵尸进程没有可执行的代码，也不能被调度，但内核仍保留其某些状态。

当一个父进程调用wait()函数时，该进程会被挂起（阻塞状态），直到捕获到子进程的退出信息（子进程变为僵尸进程），该函数才会回收子进程资源并返回（子进程不再是僵尸进程）。

4. 请简述信号在Linux系统中的作用。

信号是一种进程通讯机制，信号是发送给进程的特殊异步消息，当进程接收到信息时立即处理（不需要完成当前函数调用，甚至是当前代码行）。其本质上是软件层次上对中断机制的一种模拟，用于提醒进程某件事情已经发生。

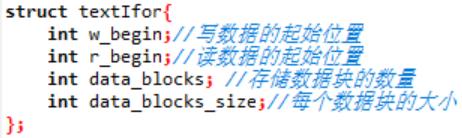
5. 请简述信号什么时候处于未决状态，并简述信号存在未决状态的作用。

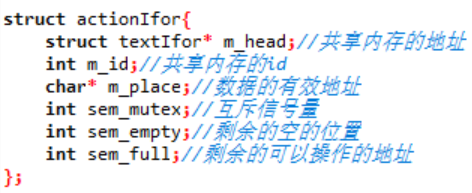
Linux系统中的信号可能处于5种状态：发送状态、阻塞状态、未决状态、递达状态和处理状态。当发送的信号被阻塞，无法到达进程，内核就会将该信号的状态设置为未决。

信号存在未决状态可以防止信号打断一些紧急的操作，保证了一些连续操作的完整性。同时可以防止信号丢失，只是暂时没有执行。

6. 请设计一种通过信号量来实现共享内存读写操作同步的方式，文字阐述即可，不需要代码实现。（提示：在写进程操作未完成时，需要防止其他进程从共享内存中读取数据）

定义两个结构textIfor和actionIfor，其中textIfor中主要存放读写数据的起始位置、存储数据块的数量和每个数据块的大小，actionIfor中主要存放共享内存的地址（类型为struct textIfor\*）、共享内存的id、数据的有效地址、互斥信号量、剩余的空的位置和剩余的可以操作的地址，其中互斥信号量、剩余的空的位置、剩余的可以操作的地址三个参数用于解决互斥和同步的信号量。





然后调用fork()函数获取key的值，将共享内存与进程绑定，创建、设置信号量并获取信号量。进行读写操作：结构textIfor类型的共享内存的地址放在共享内存的头部，数据从m\_place（m\_place==m\_head+1）处写入，每次写入更新w\_begin；数据从m\_place开始读出，读出后更新r\_begin。结构actionIfor的大小为textIfor大小+ data\_blocks \* data\_blocks\_size，然后一步步对变量和信号进行初始化。对信号量进行PV操作时，对于写数据，需要先P保证有地方可以存放数据，其次才可以进行P保证互斥性。存放数据完成后进行V取数据，在进行取数据操作时仍要保证有数据可取，其次才可以进行V。最后执行完成后释放信号量，解除进程与共享内存空间的绑定。